

# Design Optimization: An Overview

Todd Munson and Stefan Wild  
Argonne National Laboratory

# Toolkit for Advanced Optimization



$$\min \left\{ f(x) : l \leq c(x) \leq u \right\}$$

- Open-source package for numerical optimization on high-performance architectures
  - Some development funded under SciDAC-1 and SciDAC-2
  - Built on the PETSc infrastructure and linear solvers
  - Has been used in many applications since initial release in 2000 (over 6,700 downloads)
    - Chemistry (NWChem), Nuclear Physics (UNEDF – HFBTHO, HFODD, MFDn)
    - Image processing, Machine learning, Medical applications
- Support for a range of optimization types with varying derivative requirements
  - Unconstrained and bound-constrained optimization methods
    - Black-box methods – POUNDER and others
    - Gray-box methods – POUNDERs for nonlinear least squares
    - Gradient-based methods – quasi-Newton methods
    - Hessian-based methods – Newton with a trust region and/or line search
  - PDE-constrained optimization methods using derivatives
    - Linearly constrained augmented Lagrangian method
  - Methods for complementarity problems and variational inequalities
    - Active-set methods
    - Semismooth Newton methods

<http://www.mcs.anl.gov/tao>



# PDE-Constrained Optimization

$$\min \left\{ f(x, y) : c(x, y) = 0 \right\}$$

- Simulation  $c$  uniquely determines state variables  $x$  given decision variables  $y$
- Goal is to determine decision variables to optimize some metric
- Many ways to solve problems depending on available information
  - Nonlinear elimination of simulation constraint
    - Derivative-free optimization applicable when there is a small number of decision variables
    - Gradient-based methods used when adjoint information is available
  - Newton-based schemes
    - Linearized constraints
    - Quadratic approximation of objective function
    - Solve a linearly-constrained quadratic program
- Important questions that need to be answered:
  - What is the objective function? Is it smooth?
  - What are the design variables and how many? Are there discrete choices?
  - What are the constraints and how many? Are there design and state constraints?
  - Are global solutions important? Is the problem convex?
  - How much derivative information is available?



# Three Views of the Design Optimization Problem

- Unconstrained and bound-constrained optimization
  - Combine optimization criteria into a single objective
  - Eliminate the simulation constraint and solve reduced problem
  - Apply derivative-free or gradient-based methods
  - Additional constraints can be imposed
    - Analytic constraints with full information
    - Simulation constraints with partial derivative information
- Constrained optimization
  - Produce a single objective and possibly restrict feasible region
  - Apply a constrained optimization method to the full problem
  - Use Newton-based methods
- Multi-objective optimization
  - Construct Pareto surfaces
  - Explore surfaces to make tradeoffs



# Derivative-free Design Optimization

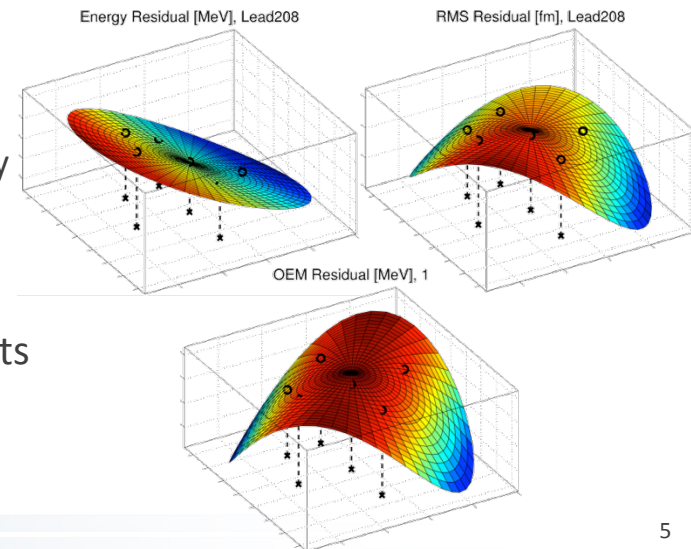
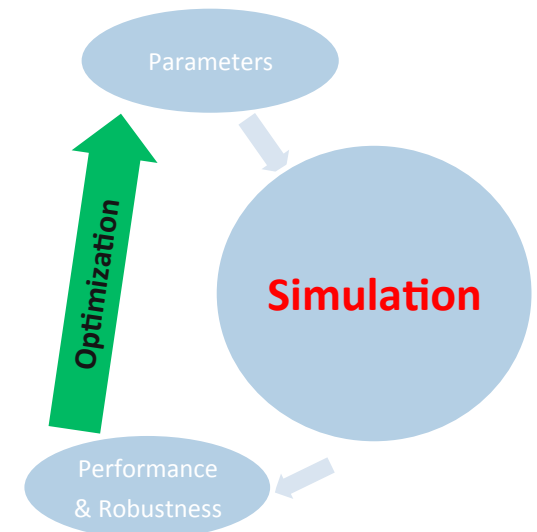
## *Simultaneous Objectives*

**Maximize performance while**

**Avoiding potential disruptions due to instabilities**

Simulation measures distances to instabilities

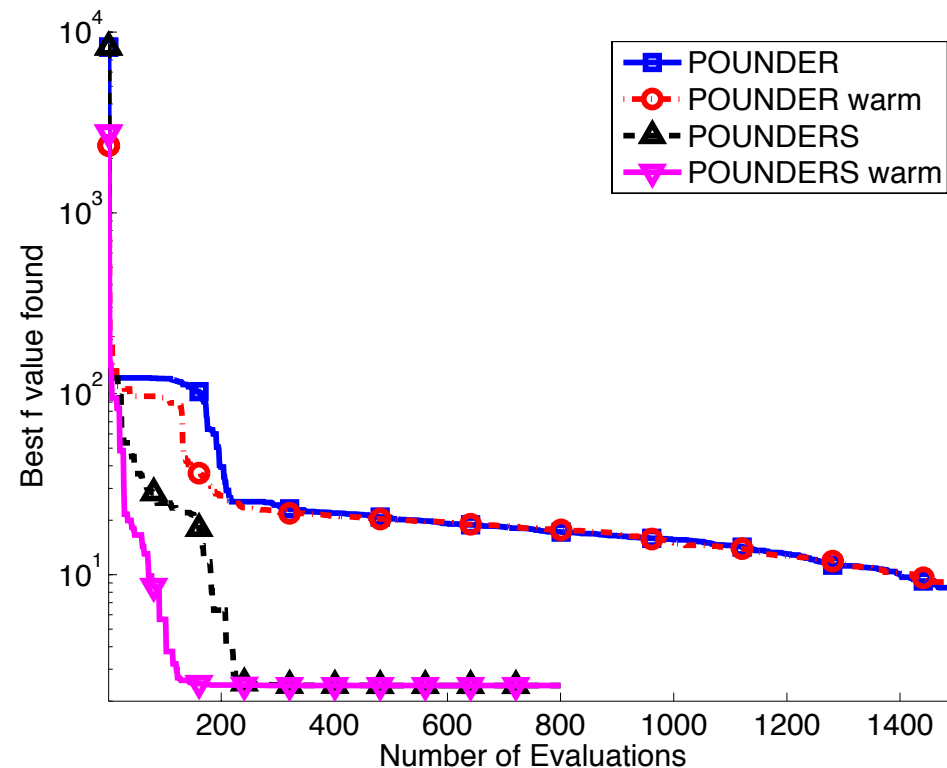
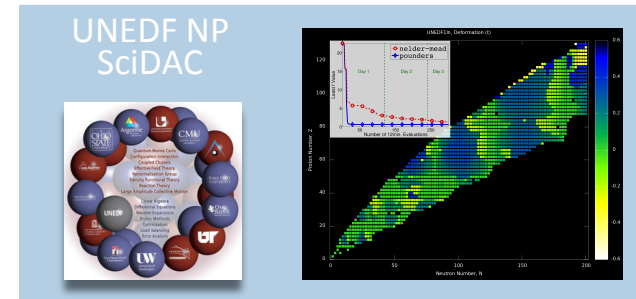
- Model-based optimization methods use past simulation results to reduce the number of simulations needed
  - Surrogate models constructed using existing **expensive** objective values
  - Optimization over surrogate models determines next parameter set to evaluate
    - Distance to an instability region can be well-behaved even though transitions are sharp
- Decisions need to be made regarding the optimization problem
  - Minimize the sum of weighted measures
  - Minimize performance with additional constraints
    - Analytic constraints on actuators, such as  $\beta_{95} \geq 3$
    - Simulation constraints like bounds on distance to instability
  - Multi-objective optimization and Pareto surfaces
- Additional information can improve performance
  - Exploit structure in the objective function and constraints
  - Use any available partial derivative information



# A Common Structure: Weighted Summations

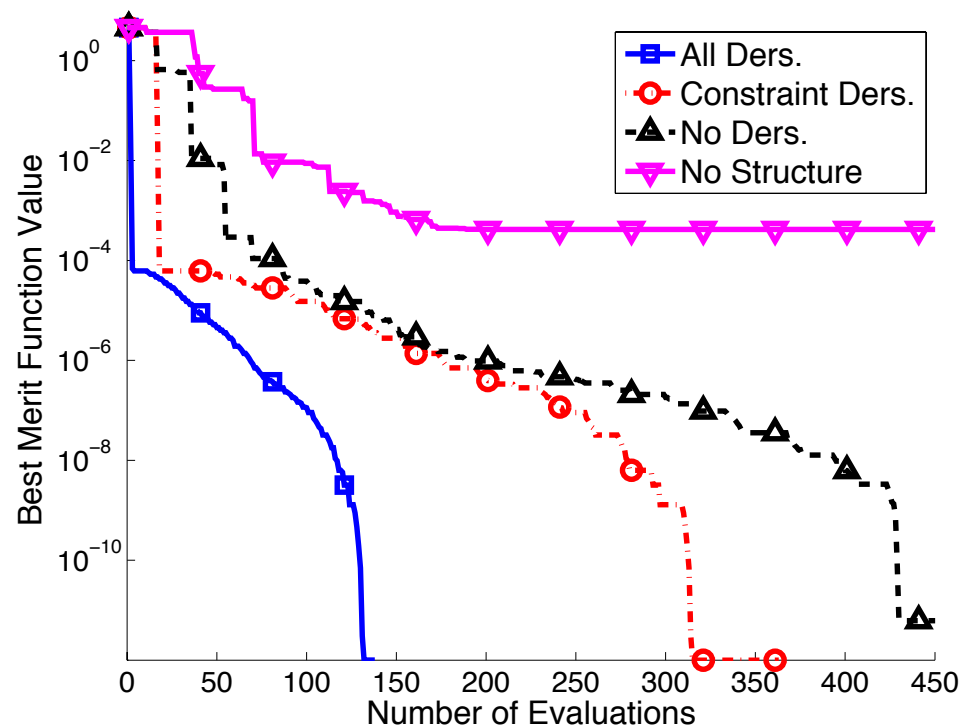
$$\max f(x) + \sum_i w_i g_i(x)$$

- Approximate the lowest level functions
  - Performance metric
  - Measures of distance to instability
  - Distance to an instability region can be well-behaved even though transitions are sharp
- Use available derivative information
  - Build better models with curvature
  - Can use partial derivatives
- Many benefits from using structure
  - Better approximations
  - Reduced number of simulation
- Objective weights are parameters
  - Explore alternative choices
    - Reuse function approximations
    - Reoptimize in fewer evaluations



# Derivative-free Constrained Optimization

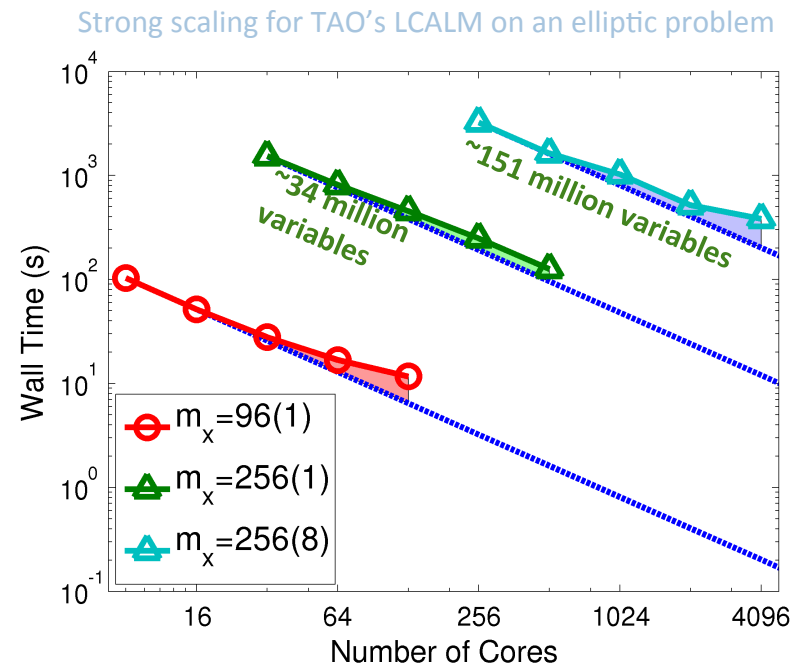
- Two types of constraints
  - Analytic constraints such as bounds on variables and linear combinations
    - For example,  $B_{95} \leq 3$
  - Simulation constraints such as bounds on the distance to outer edge instability
    - For example,  $g_1(x) \geq 0.05$
- More information on the constraints and objective functions leads to better performance
  - Structure of the functions
  - Partial or full derivative information
- Solver capability should handle any additional information provided
- Reoptimize if the functions change



Test problem ( $n=15$ ): 11 smooth constraints

# Constrained Optimization with Derivatives

- Assume availability of derivative information
  - Jacobian of constraints with respect to design and state variables
  - Gradient of the objective function with respect to design and state variables
  - Approximation or exact Hessian of the Lagrangian
- Apply a Newton-based method
  - Solve a quadratic program to obtain a search direction
  - Use a line-search or trust-region method for global convergence
- Linearly-constrained augmented Lagrangian method in TAO
  - Compute a step toward feasibility using Newton step
  - Make a step toward optimality using a reduced-space quadratic programming step
  - Perform a line-search in the full space
  - Requires two linearized forward and adjoint solves per iteration
  - Obtains good scalability on test problems



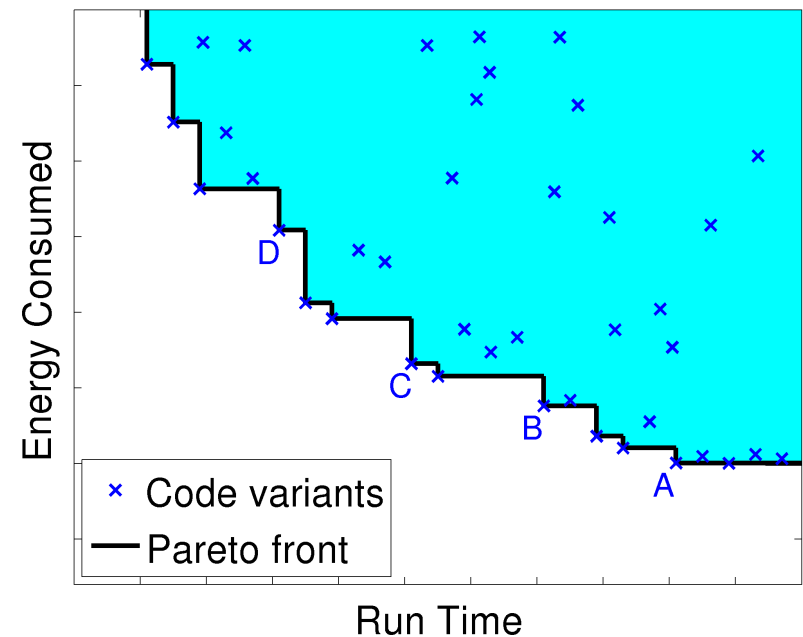
# A Digression into Nonsmooth Optimization

- Free boundaries model the interface between different physics
  - For example, the Grad-Shafranov equation partitions space into plasma and vacuum
  - Solution is continuous but nondifferentiable across the interface
  - Can be modeled with differential variational inequalities
  - Results in nonsmooth functions
- Optimization algorithms may need to change for nondifferentiable functions
  - Construct smooth models of nonsmooth functions and apply existing methods
    - Many times ignoring nonsmoothness will work
    - Can get “trapped” at points of nondifferentiability
  - Alternative methods can overcome these problems
    - There are a variety of known methods
    - Have similarities to our existing methods



# Multi-Objective Optimization

- Construct a Pareto surface
  - Explore surface to make tradeoffs
  - Useful mainly for small number of objective functions
- Performance optimization
  - Minimize run time
  - Minimize energy consumed
  - Discrete decision variables
- SUPER SciDAC Institute funds work for black-box simulations
- Similar techniques could be developed for grey-box simulations
  - Opportunity for deeper development and interactions
  - Might be applicable to design optimization problems



# Toolkit for Advanced Optimization



$$\min\{f(x) : l \leq c(x) \leq u\}$$

- Open-source package for numerical optimization on high-performance architectures
  - Some development funded under SciDAC-1 and SciDAC-2
  - Built on the PETSc infrastructure and linear solvers
  - Has been used in many applications since initial release in 2000 (over 6,700 downloads)
    - Chemistry (NWChem), Nuclear Physics (UNEDF – HFBTHO, HFODD, MFDn)
    - Image processing, Machine learning, Medical applications
- Support for a range of optimization types with varying derivative requirements
  - Unconstrained and bound-constrained optimization methods
    - Black-box methods – POUNDER and others
    - Gray-box methods – POUNDERs for nonlinear least squares
    - Gradient-based methods – quasi-Newton methods
    - Hessian-based methods – Newton with a trust region and/or line search
  - PDE-constrained optimization methods using derivatives
    - Linearly constrained augmented Lagrangian method
  - Methods for complementarity problems and variational inequalities
    - Active-set methods
    - Semismooth Newton methods

<http://www.mcs.anl.gov/tao>



# Some Discussion Questions

- What optimization problems does COMPASS need to solve?
  - What methods are you using and are you happy with them?
  - What are the bottlenecks in solving these problems?
- What new classes of problems would you want to solve with better tools?

